(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 534 583 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
07.04.1999 Bulletin 1999/14

(51) Int Cl.[6]: **G01R 21/133**, G01R 11/58

(21) Application number: **92300613.4**

(22) Date of filing: **24.01.1992**

(54) Convertible energy meter

Umwandelbarer Energiezähler

Compteur convertible d'énergie

(84) Designated Contracting States:
CH DE FR GB LI

(30) Priority: **24.09.1991 US 764863**

(43) Date of publication of application:
31.03.1993 Bulletin 1993/13

(73) Proprietor: **GENERAL ELECTRIC COMPANY**
Schenectady, NY 12345 (US)

(72) Inventors:
- **Atherton, Kenneth William**
  Saco, Maine 04072 (US)
- **Grogan, Kevin Paul**
  South Berwick, Maine 03908 (US)
- **Dastous, Susan Doris**
  Milford, New Hampshire 03055 (US)
- **Lavoie, Gregory Paul**
  Newmarket, New Hampshire 03857 (US)
- **Mancuso, Marjorie Jo**
  Exeter, New Hampshire 03833 (US)
- **Plis, Mark Joseph**
  Barrington, New Hampshire (US)

(74) Representative: **Goode, Ian Roy et al**
London Patent Operation
General Electric International, Inc.
Essex House
12-13 Essex Street
London WC2R 3AA (GB)

(56) References cited:
DE-A- 2 630 959
DE-A- 2 944 660
DE-B- 1 938 728
DE-B- 1 939 165

EP 0 534 583 B1

Printed by Jouve, 75001 PARIS (FR)

Description

[0001] A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the fascimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

I. Background of the Invention

A. Field of the Invention

[0002] The present invention generally relates to energy meters, and more specifically, relates to energy meters having the flexibility to perform various types of metering functions.

B. Related Art

[0003] In the art of energy metering, those skilled in the art typically view a meter as being composed of two (2) basic components - a metering component and a register component. The metering component is coupled to a power distribution network and generates signals indicative of power consumption. These signals are passed to the register component which performs such functions as storing the signals in a time-of-use format or a demand-type format.

[0004] Once a month or so, a meter reader is dispatched to "read" the meter. In some cases, reading a meter may simply involve downloading the register component memory into a portable electronic memory storage device carried by the meter reader.

[0005] Recently, meter manufacturers have begun manufacturing the metering component and register component in fully solid state form. This transition is due, at least in part, to the desire of power utility companies to have energy meters with the flexibility to perform a wide array of functions. Such flexibility can only be provided using solid state technology. An example of such an electrical energy meter, using a solid state measuring unit, is described in GB-A-1,575,148.

[0006] One significant and potential cost-saving feature made possible with the transition to solid state technology is to provide "convertible" meters. This means that the register component can be converted from performing, for example, demand-only functions to demand and time-of-use functions. The register component, therefore, is convertible from one function to other functions.

[0007] Until now, however, such convertibility was only provided by removing the register component and substituting another register component capable of performing the desired function. For example, a demand-only register component would be removed from the meter and replaced with a demand and time-of-use register component. Rather than being "convertible", these register components actually are "replaceable" modules. An example of a modular register component is set forth in U.S. Patent No. 5,014,213.

[0008] It is desirable, therefore, to provide a truly convertible register which does not require removing and replacing one register component for another in order to change register function.

II. Summary of the Invention

[0009] The present invention provides a programmable solid state electronic register component for an electrical energy meter, as claimed in claim 1 of the accompanying claims. In one embodiment, an operating system is utilized in register component applications. The operating system coordinates the task execution sequence and priority of major tasks to be executed by the register component. The operating system, in the one embodiment, is table driven. The tables are a form of indirect addressing to respective software routines which control operation of the register component to perform respective tasks in a predetermined order.

[0010] The tables are executed from RAM, and may point to software routines stored in either ROM or RAM. More particularly, the task tables are composed of pointers to modules which perform respective tasks such as keeping track of timers and counters for "time and date" information. By changing the operation mode of the operating system, a different set of tasks are executed.

[0011] In the normal execution mode, the operating system executes an infinite loop, i.e., unless an interrupt or "new priority" flag is set, the operating system continues to execute through the loop. This loop is sometimes referred to herein as the "kernel" loop. At the beginning of the kernel loop, a MODE byte is read to index an entry in a MODE TABLE. Each of the entries in the MODE TABLE is a pointer to a task table entry. Each entry in the task table is a pointer to a software routine.

[0012] The operating system causes the first task of the selected task table to be executed, then checks whether

there is any request for an immediate mode change. As long as there is no request for an immediate mode change, the tasks of the selected task table are executed in sequence as defined in the selected task table and until the end of the selected task table is reached, i.e., until the last task in the selected task table is executed. When the end of the selected task table is reached, or if an immediate mode change is requested, the operating system re-loads the value of the MODE byte and re-indexes the MODE TABLE. The newly selected task table is then executed.

[0013] If no mode change has been requested, either immediate or regular, then the task table "pointed to" will be the same task table just executed. An important point to note is that when a mode change occurs, task 0 of the selected task table is the first task executed. If no mode change has occurred, and the task table just executed is again selected, task 1 of the task table is the first task executed.

[0014] Task 0 of each task table is referred to as the mode initialization task and provides a place where routines may be located which only need to be executed once per mode, rather than once per task table cycle during the mode. If there are no routines in a mode that have this requirement, the mode initialization task simply returns control to the operating system, which then continues execution of the task table beginning with task 1.

[0015] In the one embodiment, to convert a register component having the present operating system from a Demand-Only register to a Time-of-Use and Demand register, for example, task tables are loaded into the register component and a different mode is selected. The additional task tables are utilized for performing necessary time-of-use functions. By selecting an appropriate mode, the execution sequence of task tables is changed so that the appropriate task tables are executed.

[0016] The present invention, importantly, provides that a register component can be changed from a demand-only register to a time-of-use and demand register without requiring that the register component be removed and replaced. The present invention facilitates providing a truly convertible register.

III. Brief Description of the Drawings

[0017] These and other objects of the present invention, together with further features and advantages thereof, will become apparent from the following detailed specification when read together with the accompanying drawings, in which:

Figure 1 is a block diagram description of one embodiment of metering means which may be utilized with the present invention;
Figure 2 is a block diagram description of one embodiment of register means;
Figure 3 is a flow chart illustrating one embodiment of a sequence of process steps in accordance with the present invention; and
Figure 4 illustrates a mode table and task tables.

IV. Detailed Description of the Drawings

[0018] Figure 1 is a block diagram description of one embodiment of metering means 100 which may be utilized with the present invention. Particularly, line current inputs and line voltage inputs are provided to the metering means 100 and watthour output pulses and VAR/Q hour output pulses are output by the metering means 100. The watthour output pulses and the VAR/Q hour pulses, respectively, are proportional to real and reactive energy consumption, respectively.

[0019] The line voltage and the line current inputs are electrically isolated and precisely scaled by respective scaling and isolation means 102 and 104 to provide secondary signals compatible with electronic circuitry. Voltage scaling to provide 1.6 Vrms at rated input voltage (120 V, for example) is suitable. Current scaling ratios of 100,000-to-one and 10,000-to-one, respectively, may be selected for self-contained (Full-scale 200 amperes) and transformer-rated (Full-scale 20 amperes) meters, respectively. Full scale secondary current from the current sensor is then 2.0 milliamps rms. A precision current-to-voltage converter 106 in the current input path converts a current signal into a voltage for compatibility with an A/D (analog-to-digital) converter. The scaling of the current-to-voltage converter is one volt output per milliamp input (1,000 ohms).

[0020] Adjustments for gain are provided by adjusting a gain stage 108 in the current path. A first A/D converter 110 is provided for the current signal and a second A/D converter 112 is provided for the voltage signal. Both A/D converters have a full-scale range of approximately +/- 3.45 volts dc, as determined by VREF from a precision voltage reference 113. The precision time base (CLOCK) 114 establishes a constant sample rate at which the A/D converters simultaneously "sample" the current and voltage inputs and convert their amplitudes to binary words. Sample rates in excess of several kilohertz are required to obtain good performance for harmonics in the input signals. Phase adjustment, to precisely match the phase of the current and voltage input signals, is provided by shifting the sample time of the voltage converter relative to the current converter in small discrete steps.

[0021] To obtain output pulses proportional to watthours, each binary-coded current sample is multiplied at a multiplier

116 by its corresponding voltage sample and the product is added to an accumulator 118. Each time the accumulated sum reaches a threshold value, proportional to the meter watthour constant, an output pulse is generated. The output pulse rate has been selected to be twelve times the rate of one disk revolution for an equivalent electromechanical meter to maintain compatibility with past generations of metering devices.

[0022] Output pulses proportional to varhours or Qhours are obtained in the same fashion as watthours except the voltage samples used are delayed, by a delay unit 120, a time equivalent to 90 degrees for varhours or 60 degrees for Qhours, either of which can be selected. Each binary coded current sample is multiplied at a multiplier 122 by the corresponding voltage sample and a separate accumulator 124 is used for varhour or Qhour accumulations. The same threshold as the watthour accumulator threshold is used as the varhour or Qhour thresholds. A typical threshold value is $144(10^{-6})$ volt-ampere-seconds (for a self-contained one-element meter for two-wire-single-phase applications).

[0023] For multiple phase loads, a multiplexer (not shown) can be used to extend the meter function. Separate isolation and scaling should be provided for each current and voltage input, but the remaining elements can be "time-shared".

[0024] Accuracy is primarily limited by noise, A/D converter resolution, linearity of input scaling, linearity of the current-to-voltage converter, and linearity of the A/D converters. Stability of performance with time and temperature is limited by the stability of input scaling, stability of the current-to-voltage converter resistor, and stability of the time base and voltage reference. An apparent inaccuracy can appear at high output pulse rates combined with a few output pulses per measurement. This occurs because of the cyclic nature of power flow at twice the line frequency causing "jitter" on the output pulse rate. This can be overcome by increasing the number of output pulses per measurement.

[0025] Figure 2 is a block diagram description of the register component means 150. The block diagram functions can be provided, for example, on a 1.2 micron CMOS application specific integrated circuit (ASIC) as is well known in the art.

[0026] The ASIC includes a microcontroller 152 coupled to an address, data, control, and umacro bus 154 (hereinafter referred to as the "bus"). A liquid crystal display controller 156 is coupled directly to the bus 154 and indirectly coupled to the bus 154 through a logic interface 158. A power outage timer 160, coupled to an oscillator 162, is coupled to the bus 154. A memory 164 including, for example, a ROM, EEPROM, and SRAM type memory blocks, also is coupled to the bus 154. The ASIC 150 also includes (but not shown) address decoding logic to define ASIC memory map, bank switching logic to extend processor address space, and communication gating logic to route register data for programming and retrieval purposes.

[0027] In operation, the microcontroller 152 performs calculating and ASIC control functions. The liquid crystal display controller 156 is utilized to control input/output means such as a liquid crystal display. The power outage timer 160, coupled to the oscillator 162, is used for timekeeping in the event of a power outage. Metering data, programs, and other information are stored in the memory 164. Further details with regard to measuring energy consumption are provided in EP-A-0,510,956, which discloses a method of operating a meter to perform the steps of generating a measure of real energy consumed, generating a measure of reactive energy consumed, and generating, from the real energy and reactive energy measures, a measure of apparent energy consumed.

[0028] Figure 3 is a flow chart 200 illustrating one embodiment of a sequence of process steps in accordance with the present invention. More particularly, one aspect of the present invention is referred to as an operating system kernel (os_kernel). The operating system controls operations of the hardware configuration hereinbefore described and the term "kernel" refers to the portion of the program which controls sequencing through respective task tables. The flow chart 200 illustrates the algorithm for executing respective task tables.

[0029] More particularly, once the os_kernel is initiated as indicated at a start block 202, the next step 204 is set the task_table_index to equal zero. Then, as indicated at a step 206, the next operation is to set the task_table equal to the table at the address identified in the mode_table indexed by the current_os_mode. More particularly, the mode byte is read to index an entry in the mode table. Each of the entries in the mode table is a pointer to a task table. For example, referring to Figure 4, in the mode table 300, the first entry is a pointer to task table 302. A subsequent entry is a pointer to task table 304. Each entry in the task table is a pointer to a main subroutine of a high-level task.

[0030] For the first iteration, the os_kernel then executes the first task of the task table. It is then determined whether the end of the task_table has been reached as indicated at a step 208. This means that it is determined whether the last task to be executed In a task table has been executed. If the last task has been executed, then operations proceed to step 210 wherein the previous_os_mode is set to the current_os_mode and the current_os_mode is set to the next_os_mode as indicated at step 212. This means that, at least with regard to steps 210 and 212, the mode of operation has not been changed and the same task tables are to be executed.

[0031] Then, at step 214, it is determined whether a mode change has been made. If a mode change has been made, then the task_table_index is set to equal zero so that the initial task in the task table to be executed is executed first. However, if no mode change has occurred, then the task_table_index is set to equal the FIRST_REPEATED_TASK. This means that if a task table is to be newly executed for the first time, then the mode initialization task (index = 0) is to be executed (step 216). If, however, the mode has not changed and the task table

is to be re-executed, the mode initialization task is not to be re-executed (step 218). That is, mode initialization tasks are to be performed only once per mode rather than once per task table cycle during the mode. Once the task table index is set to the appropriate setting, operations return to step 206.

[0032] If the end of the task_table has not been reached, as indicated at the step 208, then operations proceed to step 220. At step 220, it is determined whether an immediate mode change has been requested. If an immediate mode change has been requested, then operations proceed to step 210 and the operations hereinbefore described are executed.

[0033] If, however, no immediate mode change has been requested, then the task_page is set to equal the RAM page of the task specified in the task_table, i.e., the next task is selected for execution. The task_address is set to equal the address of the task specified in the task_table as indicated at a step 224.

[0034] If the task_address designated is in the code downloaded into RAM as indicated at step 226, then operations proceed to step 228 where the code_ram_page is set to equal the task_page. If the task_address is not in the code downloaded into RAM, however, then the data_ram_page is set to equal the task_page as indicated at step 230.

[0035] The next step 232 is to then call the routine at the task_address and to execute the next task. The task_table_index is then incremented. Operations then return to step 206.

[0036] The present invention resides in the present algorithm for an operating system kernel for use in an electricity meter. The particular modes of operations and tasks may be defined by a system operator in a manner consistent with how the meter is to be used. For example, a variety of modes such as time-of-use mode, demand only mode, fail safe mode, catch-up (after a failure) mode, and communications mode can be implemented.

[0037] As hereinbefore described, when a mode is selected, task tables may be downloaded from the ROM or EEPROM into the RAM memory block. Execution of the task tables may be performed from the RAM memory block only or from some combination of respective memory blocks. When a mode change occurs, different task tables are to be executed and may be downloaded into the RAM to replace the task tables from the previously selected mode. The operating system kernel hereinbefore described, however, does not change when a different mode is selected.

[0038] Appendix A is a listing of pseudo code for execution of an embodiment of the operating system kernel. Appendix B is a listing of the demand only mode table. Appendix C is a listing of demand only task tables. Appendix D is a listing of the time-of-use and demand mode table, and Appendix E is a listing of the task tables for time-of-use operation. These pseudo code listings are provided to further exemplify one embodiment of a task table driven operating system for an electricity meter.

APPENDIX A


```
task_table_index = INITIALIZATION_TASK
REPEAT
  WHILE (task_address <> SENTINEL_VALUE)
  AND (os_mode_change_status <> IMMEDIATE)
    task_page = page specified in task_table_buffer
    [task_table_index]
    task_address = address specified in task_table_buffer
    [task_table_index]
    IF task_address >= DATA_RAM_START
      perform change_data_ram_page(task_page)
    ELSE
      perform change_code_ram_page(task_page)
    ENDIF
    perform call to task_address
    increment task_table_index
  ENDWHILE
  previous_os_mode = current_os_mode
  current_os_mode = next_os_mode
  IF current_os_mode = previous_os_mode
    task_table_index = FIRST_REPEATED_TASK
  ELSE
    task_table_index = INITIALIZATION_TASK
    perform update_os_mode()
  ENDIF
FOREVER
```

APPENDIX B


```
rom_mode_table:
                DB      PAGE_1
                DW      rom_initialization_task_table
                DB      PAGE_1
                DW      rom_demand_only_task_table
                DB      PAGE_1
                DW      rom_demand_only_power_fail_task_table
                DB      PAGE_1
                DW      rom_demand_only_test_mode_task_table
                DB      PAGE_1
                DW      rom_optocom_mode_task_table
                DB      PAGE_1
                DW      rom_std_protocol_mode_task_table
                DB      PAGE_1
                DW      rom_fail_safe_mode_task_table
                DB      PAGE_1
                DW      rom_fail_safe_test_mode_task_table
                DB      PAGE_1
                DW      rom_fail_safe_initialization_mode_task_table
                DB      PAGE_1
                DW      rom_manufacturing_test_mode_task_table
```

APPENDIX C


```
rom_task_tables:

rom_initialization_task_table:
                DB          PAGE_0
                DW          initialization_mode_init
                DB          PAGE_0
                DW          communication_task_initialization
                DB          PAGE_0
                DW          init_parameter_program
                DB          PAGE_0
                DW          init_totals_task
                DB          PAGE_0
                DW          time_date_initialization
                DB          PAGE_0
                DW          demand_calc_initialization
                DB          PAGE_0
                DW          init_external_control_task
                DB          PAGE_0
                DW          init_check_thresholds_task
                DB          PAGE_0
                DW          init_reset_switch_control_task
                DB          PAGE_0
                DW          display_task_init
                DB          PAGE_0
                DW          init_test_task
                DB          PAGE_0
                DW          self_test_high_level_init
                DB          PAGE_0
                DW          high_level_initialization
                DB          00H
                DW          SENTINEL_VALUE

rom_demand_only_task_table:
                DB          PAGE_0
                DW          demand_only_mode_init
                DB          PAGE_0
                DW          allow_power_fail_interrupt
                DB          PAGE_0
                DW          communication_monitor
                DB          PAGE_0
                DW          prevent_power_fail_interrupt
                DB          PAGE_0
```

```
                DW      totals_task
                DB      PAGE_0
                DW      demand_only_time_date_task_support
                DB      PAGE_0
                DW      demand_only_demand_calculation
                DB      PAGE_0
                DW      allow_power_fail_interrupt
                DB      PAGE_0
                DW      prevent_power_fail_interrupt
                DB      PAGE_0
                DW      external_control_task
                DB      PAGE_0
                DW      check_thresholds_task
                DB      PAGE_0
                DW      reset_switch_control_task
                DB      PAGE_0
                DW      allow_power_fail_interrupt
                DB      PAGE_0
                DW      display_task
                DB      PAGE_0
                DW      inactive_test_mode
                DB      PAGE_0
                DW      param_pgm_task
                DB      PAGE_0
                DW      prevent_power_fail_interrupt
                DB      PAGE_0
                DW      self_test_task
                DB      PAGE_0
                DW      allow_power_fail_interrupt
                DB      00H
                DW      SENTINEL_VALUE

_demand_only_power_fail_task_table:
                DB      PAGE_0
                DW      demand_only_power_fail_mode_init
                DB      PAGE_0
                DW      demand_only_power_fail_totals
                DB      PAGE_0
                DW      demand_only_time_date_task_support
                DB      PAGE_0
                DW      demand_only_demand_calculation
                DB      00H
                DW      SENTINEL_VALUE
```

```
rom_demand_only_test_mode_task_table:
                DB      PAGE_0
                DW      test_mode_initialization_task
                DB      PAGE_0
                DW      prevent_power_fail_interrupt
                DB      PAGE_0
                DW      totals_test_support
                DB      PAGE_0
                DW      demand_only_time_date_task_support
                DB      PAGE_0
                DW      allow_power_fail_interrupt
                DB      PAGE_0
                DW      test_mode_demand_calculation
                DB      PAGE_0
                DW      external_control_task
                DB      PAGE_0
                DW      check_thresholds_task
                DB      PAGE_0
                DW      reset_switch_control_task
                DB      PAGE_0
                DW      display_task
                DB      PAGE_0
                DW      param_pgm_task
                DB      PAGE_0
                DW      active_test_mode
                DB      00H
                DW      SENTINEL_VALUE

rom_optocom_mode_task_table:
                DB      PAGE_0
                DW      optocom_mode_init
                DB      PAGE_0
                DW      optocom_message_handler
                DB      PAGE_0
                DW      param_pgm_optocom
                DB      00H
                DW      SENTINEL_VALUE

rom_std_protocol_mode_task_table:
                DB      PAGE_0
                DW      std_protocol_mode_init
                DB      PAGE_0
                DW      std_protocol_task
                DB      PAGE_0
                DW      param_pgm_optocom
                DB      00H
                DW      SENTINEL_VALUE
```

```
rom_fail_safe_mode_task_table:
                DB      PAGE_0
                DW      fail_safe_mode_init
                DB      PAGE_0
                DW      allow_power_fail_interrupt
                DB      PAGE_0
                DW      communication_monitor
                DB      PAGE_0
                DW      prevent_power_fail_interrupt
                DB      PAGE_0
                DW      fail_safe_totals
                DB      PAGE_0
                DW      allow_power_fail_interrupt
                DB      PAGE_0
                DW      display_task
                DB      PAGE_0
                DW      inactive_test_mode
                DB      PAGE_0
                DW      param_pgm_task
                DB      PAGE_0
                DW      prevent_power_fail_interrupt
                DB      PAGE_0
                DW      self_test_task
                DB      PAGE_0
                DW      allow_power_fail_interrupt
                DB      00H
                DW      SENTINEL_VALUE

rom_fail_safe_test_mode_task_table:
                DB      PAGE_0
                DW      test_mode_initialization_task
                DB      PAGE_0
                DW      allow_power_fail_interrupt
                DB      PAGE_0
                DW      totals_test_support
                DB      PAGE_0
                DW      test_mode_demand_calculation
                DB      PAGE_0
                DW      check_thresholds_task
                DB      PAGE_0
                DW      reset_switch_control_task
                DB      PAGE_0
                DW      display_task
                DB      PAGE_0
                DW      param_pgm_task
                DB      PAGE_0
                DW      active_test_mode
                DB      00H
                DW      SENTINEL_VALUE
```

```
rom_fail_safe_initialization_mode_task_table:
                DB      PAGE_0
                DW      fail_safe_initialization_mode_init
                DB      PAGE_0
                DW      communication_task_initialization
                DB      PAGE_0
                DW      init_parameter_program
                DB      PAGE_0
                DW      init_totals_task
                DB      PAGE_0
                DW      time_date_initialization
                DB      PAGE_0
                DW      demand_calc_initialization
                DB      PAGE_0
                DW      init_check_thresholds_task
                DB      PAGE_0
                DW      init_reset_switch_control_task
                DB      PAGE_0
                DW      display_task_init
                DB      PAGE_0
                DW      init_test_task
                DB      PAGE_0
                DW      self_test_fail_safe_init
                DB      PAGE_0
                DW      high_level_initialization
                DB      00H
                DW      SENTINEL_VALUE

rom_manufacturing_test_mode_task_table:
                DB      PAGE_0
                DW      manufacturing_test_mode_init
                DB      PAGE_0
                DW      manufacturing_test_task
                DB      00H
                DW      SENTINEL_VALUE
```

APPENDIX D


```
ram_mode_table:
                DB      PAGE_1
                DW      tou_demand_initialization_task_table
                DB      PAGE_0
                DW      rom_demand_only_task_table
                DB      PAGE_0
                DW      rom_demand_only_power_fail_task_table
                DB      PAGE_0
                DW      rom_demand_only_test_mode_task_table
                DB      PAGE_0
                DW      rom_optocom_mode_task_table
                DB      PAGE_0
                DW      rom_std_protocol_mode_task_table
                DB      PAGE_0
                DW      rom_fail_safe_mode_task_table
                DB      PAGE_0
                DW      rom_fail_safe_test_mode_task_table
                DB      PAGE_0
                DW      rom_fail_safe_initialization_mode_task_table
                DB      PAGE_0
                DW      rom_manufacturing_test_mode_task_table
                DB      PAGE_1
                DW      tou_demand_task_table
                DB      PAGE_1
                DW      catch_up_mode_task_table
                DB      PAGE_1
                DW      tou_demand_test_mode_task_table
                DB      PAGE_0
                DW      0000H                   ;extra entry
                DB      PAGE_0
                DW      0000H                   ;extra entry
```

APPENDIX E


```
ram_task_tables:

tou_demand_initialization_task_table:
                DB      PAGE_0
                DW      initialization_mode_init
                DB      PAGE_0
                DW      high_level_initialization
                DB      PAGE_0
                DW      communication_task_initialization
                DB      PAGE_0
                DW      init_parameter_program
                DB      PAGE_0
                DW      init_totals_task
                DB      PAGE_0
                DW      time_date_initialization
                DB      PAGE_2
                DW      init_prog_dates_task
                DB      PAGE_0
                DW      demand_calc_initialization
                DB      PAGE_2
                DW      init_season_change_task
                DB      PAGE_0
                DW      init_external_control_task
                DB      PAGE_2
                DW      init_rate_change_task
                DB      PAGE_2
                DW      init_load_profile_task
                DB      PAGE_0
                DW      init_check_thresholds_task
                DB      PAGE_0
                DW      init_reset_switch_control_task
                DB      PAGE_0
                DW      display_task_init
                DB      PAGE_0
                DW      init_test_task
                DB      PAGE_0
                DW      self_test_high_level_init
                DB      00H
                DW      SENTINEL_VALUE
```

```
tou_demand_task_table:
                DB      PAGE_2
                DW      tou_demand_mode_init
                DB      PAGE_0
                DW      abort_after_power_fail
                DB      PAGE_0
                DW      communication_monitor
                DB      PAGE_0
                DW      continue_after_power_fail
                DB      PAGE_0
                DW      prevent_power_fail_interrupt
                DB      PAGE_0
                DW      totals_task
                DB      PAGE_0
                DW      allow_power_fail_interrupt
                DB      PAGE_2
                DW      tou_demand_time_date_task_support
                DB      PAGE_2
                DW      programmable_dates_task
                DB      PAGE_2
                DW      tou_demand_demand_calculation
                DB      PAGE_2
                DW      season_change_task
                DB      PAGE_2
                DW      rate_change_task
                DB      PAGE_2
                DW      load_profile_recording_task
                DB      PAGE_0
                DW      external_control_task
                DB      PAGE_0
                DW      check_thresholds_task
                DB      PAGE_0
                DW      reset_switch_control_task
                DB      PAGE_0
                DW      abort_after_power_fail
                DB      PAGE_0
                DW      display_task
                DB      PAGE_0
                DW      inactive_test_mode
                DB      PAGE_0
                DW      param_pgm_task
                DB      PAGE_0
                DW      continue_after_power_fail
                DB      PAGE_0
                DW      prevent_power_fail_interrupt
                DB      PAGE_0
                DW      self_test_task
                DB      PAGE_0
                DW      allow_power_fail_interrupt
                DB      00H
                DW      SENTINEL_VALUE
```

```
catch_up_mode_task_table:
                DB      PAGE_2
                DW      catch_up_mode_init
                DB      PAGE_2
                DW      prevent_power_fail_interrupt
                DB      PAGE_2
                DW      totals_catch_up_support
                DB      PAGE_0
                DW      allow_power_fail_interrupt
                DB      PAGE_2
                DW      tou_demand_time_date_task_support
                DB      PAGE_2
                DW      programmable_dates_task
                DB      PAGE_2
                DW      fast_catch_up_task
                DB      PAGE_2
                DW      tou_demand_demand_calculation
                DB      PAGE_2
                DW      season_change_task
                DB      PAGE_2
                DW      rate_change_task
                DB      PAGE_2
                DW      load_profile_recording_task
                DB      00H
                DW      SENTINEL_VALUE
```

```
tou_demand_test_mode_task_table:
                DB      PAGE_0
                DW      test_mode_initialization_task
                DB      PAGE_0
                DW      prevent_power_fail_interrupt
                DB      PAGE_0
                DW      totals_test_support
                DB      PAGE_2
                DW      tou_demand_time_date_task_support
                DB      PAGE_2
                DW      programmable_dates_task
                DB      PAGE_0
                DW      test_mode_demand_calculation
                DB      PAGE_2
                DW      season_change_task
                DB      PAGE_2
                DW      rate_change_task
                DB      PAGE_2
                DW      load_profile_recording_task
                DB      PAGE_0
                DW      external_control_task
                DB      PAGE_0
                DW      check_thresholds_task
                DB      PAGE_0
                DW      reset_switch_control_task
                DB      PAGE_0
                DW      allow_power_fail_interrupt
                DB      PAGE_0
                DW      display_task
                DB      PAGE_0
                DW      param_pgm_task
                DB      PAGE_0
                DW      active_test_mode
                DB      00H
                DW      SENTINEL_VALUE
```

Claims

1. A programmable solid-state electronic register component (150) for an electrical energy meter, comprising:

   a mode table (300);
   a plurality of task tables (302,304);
   a plurality of modules;
   wherein said mode table comprises an array of task pointers to respective ones of said plurality of task tables,
   wherein each of said plurality of task tables comprises an array of module pointers to corresponding ones of said plurality of modules,
   wherein each of said plurality of modules comprises a plurality of instructions to be executed during the performance of a corresponding task, which is included in at least one of predetermined groups of tasks, and
   wherein each of said predetermined groups of tasks is defined by one of said task tables and corresponds to a predetermined mode of operation such as a time-of-use and demand mode of operation or a demand-only mode of operation; and
   micro-controller means (152) for operating said register component in said predetermined mode of operation

by executing a corresponding predetermined group of tasks, said micro-controller means being characterised by:

means for indexing (204,206) to a task pointer in said mode table to thereby select one of said predetermined groups of tasks, said task pointer corresponding to a selected task table,

means for executing (208,220-230,234) said one of said predetermined groups of tasks by repeatedly indexing said selected task table to respective module pointers, and

means for executing (232) said plurality of modules corresponding to said respective module pointers.

2. The programmable solid-state electronic register component of claim 1, wherein said mode table said plurality of task tables and said plurality of modules are contained in memory (164) comprising read-only memory and random-access memory and wherein said means for executing said one of said predetermined groups of tasks comprises means for executing said one of said predetermined groups of tasks in said random-access memory.

3. The programmable solid-state electronic register component of claim 2, further comprising input/output means (156) responsive to actuation by a user for selecting said programmed mode of operation and means for down-loading (154) said task tables corresponding to said programmed mode of operation to said random-access memory.

4. The programmable solid-state electronic register component of claim 3, wherein said means for down-loading said task tables comprises means for downloading (154) said task tables corresponding to said programmed mode of operation from said read-only memory to said random-access memory.

5. The programmable solid-state electronic register component of claim 4, wherein said input-output means comprises a liquid crystal display.

6. The programmable solid-state electronic register component of claim 1, wherein said arrays of module pointers for each of said task tables include a mode-initialization pointer (304, Entry = 0) corresponding to a respective mode-initialization module and wherein said means for executing said plurality of modules comprises means for executing said respective mode-initialization module only once during said programmed mode of operation.

7. The programmable solid-state electronic register component of claim 3, wherein said arrays of module pointers for each of said task tables include a mode-initialization pointer (304, Entry = 0) corresponding to a respective mode-initialization module and wherein said means for executing said plurality of modules comprises means for executing (218) said respective mode-initialization module only once during said programmed mode of operation.

8. The programmable solid-state electronic register component of claim 7, further comprising:

means for halting (220) said means for executing said one of said predetermined groups of tasks prior to indexing the entire selected task table; and

means for initializing (216) said means for downloading said task tables corresponding to another programmed mode of operation from said read-only memory to said random-access memory.

9. The programmable solid-state electronic register component of claim 2, wherein said memory and said microcontroller means are contained in an application specific integrated circuit.

Patentansprüche

1. Programmierbare elektronische Festkörper-Registerkomponente (150) für ein elektrisches Energiemeßgerät, enthaltend:

eine Modus-Tabelle (300);
eine Anzahl von Aufgaben-Tabellen (302,304);
eine Anzahl von Moduln;
wobei die Modus-Tabelle ein Array von Aufgaben-Zeigern auf entsprechende der Anzahl von Aufgaben-Tabellen aufweist;
wobei jede der Anzahl von Aufgaben-Tabellen ein Array von Modul-Zeigern auf entsprechende der Anzahl von Moduln aufweist;

wobei jede der Anzahl von Moduln eine Anzahl von Befehlen aufweist, die während der Abarbeitung von einer entsprechenden Aufgabe auszuführen sind, die in wenigstens einer der vorbestimmten Gruppen von Aufgaben enthalten ist, und
wobei jede der vorbestimmten Gruppen von Aufgaben durch eine der Aufgaben-Tabellen definiert ist und einem vorbestimmten Betriebsmodus entspricht, wie beispielsweise einem Verwendungszeit- und Bedarfs-Betriebsmodus oder einem Nur-Bedarfs-Betriebsmodus; und
eine Mikroprozessor-Steuereinrichtung (152) zum Betreiben der Registerkomponente in dem vorbestimmten Betriebsmodus, indem eine entsprechende vorbestimmte Gruppe von Aufgaben ausgeführt wird, wobei die Mikroprozessor-Steuereinrichtung

gekennzeichnet ist durch:

eine Einrichtung zum Indexieren (204,206) auf einen Aufgaben-Zeiger in der Modus-Tabelle, um dadurch eine der vorbestimmten Gruppen von Aufgaben zu wählen, wobei der Aufgaben-Zeiger einer gewählten Aufgaben-Tabelle entspricht,
eine Einrichtung zum Ausführen (208,220-230,234) der einen der vorbestimmten Gruppen von Aufgaben, indem wiederholt die gewählte Aufgaben-Tabelle auf entsprechende ModulZeiger indexiert wird, und
eine Einrichtung zum Ausführen (232) der Anzahl von Moduln entsprechend den entsprechenden Modul-Zeigern.

2. Programmierbare elektronische Festkörper-Registerkomponente nach Anspruch 1, wobei die Modus-Tabelle, die Anzahl von Aufgaben-Tabellen und die Anzahl von Moduln in einem Speicher (164) enthalten sind, der einen Festwertspeicher und einen Arbeitsspeicher aufweist, und wobei die Einrichtung zum Ausführen von einer der vorbestimmten Gruppen von Aufgaben eine Einrichtung aufweist zum Ausführen der einen der vorbestimmten Gruppen von Aufgaben in dem Arbeitsspeicher.

3. Programmierbare elektronische Festkörper-Registerkomponente nach Anspruch 2, wobei ferner eine Eingabe/Ausgabe-Einrichtung (156), die auf eine Betätigung durch einen Benutzer zum Wählen des programmierten Betriebsmodus anspricht, und eine Einrichtung zum Runterladen (154) der Aufgaben-Tabellen entsprechend dem programmierten Betriebsmodus in den Arbeitsspeicher vorgesehen sind.

4. Programmierbare elektronische Festkörper-Registerkomponente nach Anspruch 3, wobei die Einrichtung zum Runterladen der Aufgaben-Tabellen eine Einrichtung zum Runterladen (154) der Aufgaben-Tabellen entsprechend dem programmierten Betriebsmodus von dem Festwertspeicher in den Arbeitsspeicher aufweist.

5. Programmierbare elektronische Festkörper-Registerkomponente nach Anspruch 4, wobei die Eingabe/Ausgabe-Einrichtung ein Flüssigkristalldisplay aufweist.

6. Programmierbare elektronische Festkörper-Registerkomponente nach Anspruch 1, wobei die Arrays von Modul-Zeigern für jede der Aufgaben-Tabellen einen Modus-Initialisierungszeiger (304, Eintrag = 0) entsprechend einem entsprechenden Modus-Initialisierungsmodus enthält, und wobei die Einrichtung zum Ausführen der Anzahl von Moduln eine Einrichtung aufweist zum Ausführen des entsprechenden Modus-Initialisierungsmoduls nur einmal während des programmierten Betriebsmodus.

7. Programmierbare elektronische Festkörper-Registerkomponente nach Anspruch 3, wobei die Arrays von Modul-Zeigern für jede der Aufgaben-Tabellen einen Modus-Initialisierungszeiger (304, Eintrag = 0) entsprechend einem entsprechenden Modus-Initialisierungsmodul aufweist, und wobei die Einrichtung zum Ausführen der Anzahl von Moduln eine Einrichtung zum Ausführen (218) des entsprechenden Modus-Initialisierungsmoduls nur einmal während des programmierten Betriebsmodus aufweist.

8. Programmierbare elektronische Festkörper-Registerkomponente nach Anspruch 7, ferner enthaltend:

eine Einrichtung zum Halten (220) der Einrichtung zum Ausführen der einen der vorbestimmten Gruppen von Aufgaben vor dem Indexieren der gesamten gewählten Aufgaben-Tabelle; und
eine Einrichtung zum Initialisieren (216) der Einrichtung zum Runterladen der Aufgaben-Tabellen entsprechend einem anderen programmierten Betriebsmodus von dem Festwertspeicher in den Arbeitsspeicher.

9. Programmierbare elektronische Festkörper-Registerkomponente nach Anspruch 2, wobei der Speicher und die

Mikroprozessor-Steuereinrichtung in einer Anwendungs-spezifischen Integrierten Schaltung enthalten sind.

Revendications

1. Composant (150) formant registre électronique programmable à semiconducteurs pour un compteur d'énergie électrique, comprenant :

   une table de mode (300);
   une pluralité de tables de tâches (302,304);
   une pluralité de modules;
   dans lequel ladite table de mode comprend un ensemble de pointeurs de tâche pointant vers les tables correspondantes de ladite pluralité de tables de tâches,
   dans lequel chacune des tables de ladite pluralité de tables de tâches comprend un ensemble de pointeurs de modules pointant vers les modules correspondants de ladite pluralité de modules,
   dans lequel chacun des modules de ladite pluralité de modules comprend une pluralité d'instructions à exécuter pendant l'accomplissement d'une tâche correspondante, qui se trouve dans au moins un groupe de tâches parmi des groupes de tâches prédéterminés, et
   dans lequel chacun desdits groupes de tâches prédéterminés est défini par une desdites tables de tâches et correspond à un mode de fonctionnement prédéterminé, comme un mode de durée d'utilisation et de puissance demandée ou un mode de puissance demandée seule; et
   des moyens formant microcontrôleur (152) et destinés à faire fonctionner ledit composant formant registre dans ledit mode de fonctionnement prédéterminé en exécutant un groupe de tâches prédéterminé correspondant, lesdits moyens formant microcontrôleur étant caractérisés par :
   des moyens servant à indexer (204,206) un pointeur de tâche dans ladite table de mode pour sélectionner un desdits groupes de tâches prédéterminés, ledit pointeur de tâche correspondant à une table de tâches sélectionnée,
   des moyens servant à exécuter (208,220-230,234) ledit groupe de tâches prédéterminé sélectionné en indexant à plusieurs reprises ladite table de tâches sélectionnée aux pointeurs de modules respectifs, et
   un moyen servant à exécuter (232) ladite pluralité de modules correspondant auxdits pointeurs de modules respectifs.

2. Composant formant registre électronique programmable à semiconducteurs selon la revendication 1, dans lequel ladite table de mode, ladite pluralité de tables de tâches et ladite pluralité de modules sont contenues dans une mémoire (164) comprenant de la mémoire morte et de la mémoire vive, et dans lequel lesdits moyens servant à exécuter ledit groupe de tâches prédéterminé sélectionné comprennent un moyen servant à exécuter ledit groupe de tâches prédéterminé sélectionné dans ladite mémoire vive.

3. Composant formant registre électronique programmable à semiconducteurs selon la revendication 2, comprenant de plus un moyen d'entrée/sortie (156) qui réagit à la commande d'un utilisateur pour choisir ledit mode de fonctionnement programmé, et un moyen servant à télécharger (154) lesdites tables de tâches correspondant audit mode de fonctionnement programmé dans ladite mémoire vive.

4. Composant formant registre électronique programmable à semiconducteurs selon la revendication 3, dans lequel ledit moyen servant à télécharger lesdites tables de tâches comprend un moyen servant à télécharger (154) lesdites tables de tâches correspondant audit mode de fonctionnement programmé de ladite mémoire morte dans ladite mémoire vive.

5. Composant formant registre électronique programmable à semiconducteurs selon la revendication 4, dans lequel ledit moyen d'entrée/sortie comprend un afficheur à cristaux liquides.

6. Composant formant registre électronique programmable à semiconducteurs selon la revendication 1, dans lequel lesdits ensembles de pointeurs de modules pour chacune desdites tables de tâches comprennent un pointeur d'initialisation de mode (304, Rang = 0) correspondant à un module d'initialisation de mode respectif et dans lequel ledit moyen servant à exécuter ladite pluralité de modules comprend un moyen servant à exécuter ledit module d'initialisation de mode respectif une seule fois pendant ledit mode de fonctionnement programmé.

7. Composant formant registre électronique programmable à semiconducteurs selon la revendication 3, dans lequel

lesdits ensembles de pointeurs de modules pour chacune desdites tables de tâches comprennent un pointeur d'initialisation de mode (304, Rang = 0) correspondant à un module d'initialisation de mode respectif et dans lequel ledit moyen servant à exécuter ladite pluralité de modules comprend un moyen servant à exécuter (218) ledit module d'initialisation de mode respectif une seule fois pendant ledit mode de fonctionnement programmé.

8. Composant formant registre électronique programmable à semiconducteurs selon la revendication 7, comprenant de plus :

un moyen permettant l'arrêt (220) desdits moyens servant à exécuter ledit groupe de tâches prédéterminé sélectionné avant l'indexage de la table de tâches sélectionnée complète; et
un moyen servant à initialiser (216) ledit moyen servant à télécharger lesdites tables de tâches correspondant à un autre mode de fonctionnement programmé de ladite mémoire morte dans ladite mémoire vive.

9. Composant formant registre électronique programmable à semiconducteurs selon la revendication 2, dans lequel ladite mémoire et lesdits moyens formant microcontrôleur sont contenus dans un circuit intégré de type ASIC.

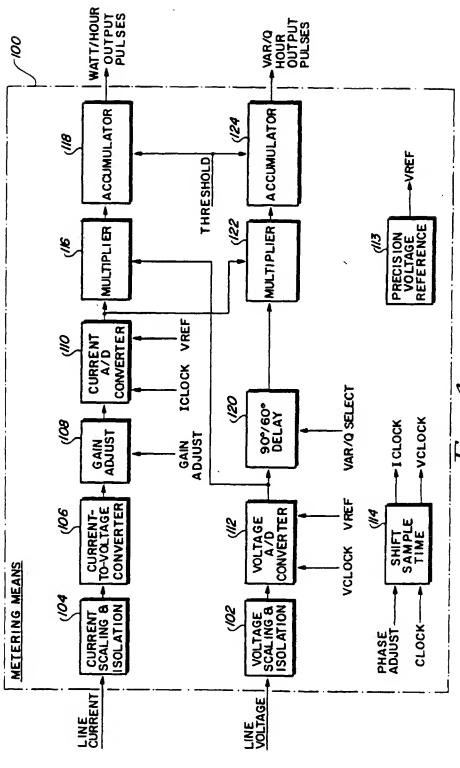
100
METERING MEANS
LINE CURRENT
LINE VOLTAGE
102
VOLTAGE SCALING & ISOLATION
104
CURRENT SCALING & ISOLATION
106
CURRENT-TO-VOLTAGE CONVERTER
108
GAIN ADJUST
110
CURRENT A/D CONVERTER
112
VOLTAGE A/D CONVERTER
114
SHIFT SAMPLE TIME
116
MULTIPLIER
118
ACCUMULATOR
120
90°/60° DELAY
122
MULTIPLIER
124
ACCUMULATOR
113
PRECISION VOLTAGE REFERENCE
PHASE ADJUST
CLOCK
VCLOCK
VREF
I CLOCK
VCLOCK
VAR/Q SELECT
GAIN ADJUST
ICLOCK
VREF
THRESHOLD
VREF
WATT/HOUR OUTPUT PULSES
VAR/Q HOUR OUTPUT PULSES

FIG. 1

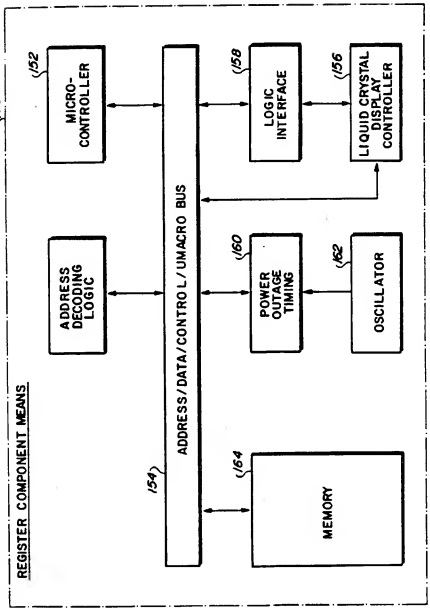
150
REGISTER COMPONENT MEANS
152
MICRO-CONTROLLER
158
LOGIC INTERFACE
156
LIQUID CRYSTAL DISPLAY CONTROLLER
ADDRESS DECODING LOGIC
154
ADDRESS/DATA/CONTROL/UMACRO BUS
160
POWER OUTAGE TIMING
162
OSCILLATOR
164
MEMORY

FIG. 2

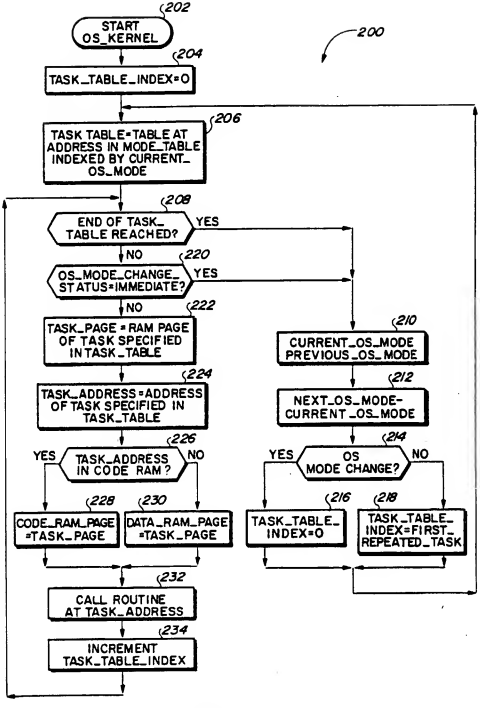
202
START OS_KERNEL
200
204
TASK_TABLE_INDEX=0
206
TASK TABLE=TABLE AT ADDRESS IN MODE_TABLE INDEXED BY CURRENT_ OS_MODE
208
END OF TASK_ TABLE REACHED?
YES
NO
220
OS_MODE_CHANGE_ STATUS=IMMEDIATE?
YES
NO
222
TASK_PAGE = RAM PAGE OF TASK SPECIFIED IN TASK_TABLE
224
TASK_ADDRESS=ADDRESS OF TASK SPECIFIED IN TASK_TABLE
226
YES
TASK_ADDRESS IN CODE RAM?
NO
228
CODE_RAM_PAGE =TASK_PAGE
230
DATA_RAM_PAGE =TASK_PAGE
210
CURRENT_OS_MODE PREVIOUS_OS_MODE
212
NEXT_OS_MODE= CURRENT_OS_MODE
214
YES
OS MODE CHANGE?
NO
216
TASK_TABLE_ INDEX=0
218
TASK_TABLE_ INDEX=FIRST_ REPEATED_TASK
232
CALL ROUTINE AT TASK_ADDRESS
234
INCREMENT TASK_TABLE_INDEX

FIG. 3

FIG. 4

TABLE	ENTRY	BYTE	VALUE
TASKTBL	0	0	MEMORY PAGE FOR MODE INITALIZATION TASK
	0	1	ADDRESS FOR MODE INITALIZATION TASK
	1	3	MEMORY PAGE FOR TASK 1
	1	4	ADDRESS FOR TASK 1
	n	3n	MEMORY PAGE FOR TASK n
	n	3n+	ADDRESS FOR TASK n
	n+1	3n+3	UNUSED
	n+1	3n+4	SENTINEL VALUE (0)

302

300

TABLE	ENTRY	BYTE	VALUE
MODETBL	0	0	MEMORY PAGE FOR TASK TABLE 1
	0	1	ADDRESS FOR TASK TABLE 1
	1	3	MEMORY PAGE FOR TASK TABLE 2
	1	4	ADDRESS FOR TASK TABLE 2
	i	3i	MEMORY PAGE FOR TASK TABLE i
	i	3i+1	ADDRESS FOR TASK TABLE i
	i+1	3i+3	UNUSED
	i+1	3i+4	SENTINEL VALUE (0)

304

TABLE	ENTRY	BYTE	VALUE
TASKTBL	0	0	MEMORY PAGE FOR MODE INITALIZATION TASK
	0	1	ADDRESS FOR MODE INITIALIZATION TASK
	1	3	MEMORY PAGE FOR TASK 1
	1	4	ADDRESS FOR TASK 1
	n	3n	MEMORY PAGE FOR TASK n
	n	3n+1	ADDRESS FOR TASK n
	n+1	3n+3	UNUSED
	n+1	3n+4	SENTINEL VALUE (0)